Universal Fuzzy System to Takagi-Sugeno Fuzzy System Compiler

Enrique Frias-Martinez
Computer Science Department, New York University
Room 719, 715 Broadway
New York, NY 10003
E-mail: frias@cs.nyu.edu

Abstract – **In this paper a compiler that transforms a universal fuzzy system into a Takagi-Sugeno fuzzy system is presented. A universal fuzzy system is defined as a fuzzy system with generic membership functions, T-norm, T-conorm, propagation operator, aggregation operator and defuzzification algorithm. The Takagi-Sugeno system obtained is an approximation of the original fuzzy system based on keeping the certainty of the designer. This means that the inputs for which the designer had a complete certainty will be approximated without error.**

I. INTRODUCTION

Fuzzy logic has been successfully introduced in a wide range of applications, ranging from classical control systems to decision support systems as nuclear plant supervision [6], medical applications [7] or automotive applications [9]. [8] and [1] present a wide variety of fuzzy logic applications.
The characteristics of a fuzzy system will depend on the application that solves. Different applications will require different T-norm, different membership functions, or different defuzzification algorithms. Not only that, but also the same problem can be solved by different designers using fuzzy systems with different characteristics (different membership functions, different inference mechanism, or different defuzzification algorithm for example).

Although this variety of fuzzy systems exists for application purposes, the formal studies of fuzzy systems are based mainly in some standard systems. One of those standard systems is the Takagi-Sugeno system.

Takagi-Sugeno (T-S) systems have been broadly studied, and almost all commercial platforms (hardware and software) are able to execute a Takagi-Sugeno system.

This paper presents an approximate compiler that transforms a universal fuzzy system into a Takagi-Sugeno fuzzy system. A universal fuzzy system is defined as a system where the membership functions, rules, T-norm, T-conorm, propagation operator, aggregation operator and defuzzification algorithm are programmable. The concept of approximate compilation is defined as maintaining the certainty of the designer, so the approximation to the original system is done only in the areas where the designer has partial certainty.

The use of a compiler that transforms a universal fuzzy system into an approximate Takagi-Sugeno system, will allow to run any fuzzy system in the set of platforms (both software and hardware) designed to execute Takagi-Sugeno systems [2][3][5].

II. INTRODUCTION TO TS SYSTEMS

Takagi-Sugeno fuzzy systems were introduced in [4]. A Takagi-Sugeno system is based on using functions as the consequent of the rules instead of membership functions as used by Mamdami systems.

A fuzzy Takagi-Sugeno system is defined by a set of rules $\{R_1,...,R_m\}$ where each rule R_i can be formulated as:

$$R_i = IF\ I_1\ is\ A_{1,j}\ and\ ...\ and\ I_n\ is\ A_{n,k}\ THEN$$
$$Y_i = p_0 + p_1 I_1 + ... + p_n I_n \quad (1)$$

Where $\{I_1,..,I_n\}$ is the set of inputs of the system, $\{A_{1,j},...,A_{n,k}\}$ the set of membership functions that define the rule, and Y_i the output of rule R_i. Typically the conclusion of the rule is a zero-order function, and the rules are expressed as:

$$R_i = IF\ I_1\ is\ A_{1,j}\ and\ ...\ and\ I_n\ is\ A_{n,k}\ THEN\ Y_i = p_0 \quad (2)$$

Given and input $\{I_1,...,I_n\}$ the output of the system will be given by the weighted averages of $\{Y_1,...,Y_m\}$:

$$Y = \frac{\sum_{i=1}^{m} w_i Y_i}{\sum_{i=1}^{m} w_i} \quad (3)$$

Where w_i is the overall degree of truth of rule R_i, and is obtained as:

$$w_i = \prod_{l=1}^{n} A_{i,l}(I_l) \quad (4)$$

In a Takagi-Sugeno system the defined T-norm is product, the propagation operator is also product, the aggregation operator is sum, and the defuzzification algorithm is weighted average. Also Typically the membership functions are triangular or trapezoidal with an overlap factor of two.

A Takagi-Sugeno fuzzy model is based on the division of the fuzzy spaces into subspaces. In each of those fuzzy subspaces the system constructs a linear input-output relation.

In the typical situation of zero-order output, triangular or trapezoidal membership functions and overlap factor of two, a Takagi-Sugeno system is equivalent to a linear interpolation [5].

0-7803-7280-8/02/$10.00 ©2002 IEEE

III. APPROXIMATE COMPILER

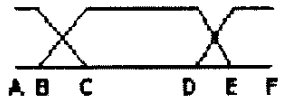
The compiler proposed is defined in two steps. The first one, from the description of a universal fuzzy system, obtains the relevant data of the system in order to maintain the certainty of the designer. The second one, from the relevant data, constructs the equivalent Takagi-Sugeno system.

The equivalent Takagi-Sugeno system is an approximation of the original system in which the zones of total certainty of the designer keep their original value, and the zones where the certainty of the designer is only partial are approximated to the original system. The error in this case is irrelevant because is given to a value which is not exact but fuzzy by construction.

A. Obtaining the relevant data from the system

FS is a vector $FS=(T_n,T_c,PO,AO,D,R,MF,MF_o,I,O)$ that describes a fuzzy system with T_n the T-norm, T_c the T-conorm, *PO* the propagation operator, *AO* the aggregation operator, *D* the defuzification algorithm, *R* the set of rules, *MF* the set of normalized and convex membership functions of the inputs, MF_o the membership functions of the output, *I* the vector of inputs and *O* the output of the system (Card(*O*)=1).

For each $I_i \in I$, the Activation Intervals of I_i, AI_i, are defined as the union of the set of intervals, left-closed and right-open, except the last one which is also right-closed, given by the extreme points of the kernel and the support of each one of the membership functions defined in I_i. AI_i is obtained from $(MF_{i,1},...,MF_{i,p})$ with $p=Card(MF_i)$. Fig. 1 presents an example of Activation Intervals.



$AI=\{[A,B),[B,C),[C,D),[D,E),[E,F]\}$

Fig. 1 Activation Intervals AI

The Activation Intervals can be obtained from the Activation Points. The Activation Points are defined as the set of points of each dimension of the system that define the kernel and support of each linguistic label. Formally, the Activation Points of I_i, AP_i can be obtained as:

$$AP_i = F_e(F_o(\cup(Supp_e(MF_{i,m}) \cup Kernel_e(MF_{i,m})))) \text{ with } m=1,...,p. \quad (5)$$

From that, the Activation Intervals, are obtained as:

$$AI_i = F_{in}(AP_i), \quad (6)$$

where $Supp_e$ and $Kernel_e$ are functions that obtain the extreme points that define the kernel and the support of a label, F_o is a function that orders a set of points, F_e is a function that eliminates the repeated points of a set, and F_{in} obtains the intervals, left-closed and right-open, that are defined by a set of points.

Given AI_i, $i=1,...,Card(I)$, and *S* the n-dimensional input space given by $(I_1,...,I_n)$, *P* is defined as a partition of *S* given by the cartesian product (×) of AI_i, $i=1,...,Card(I)$:

$$P = \times AI_i,\ i=1,...,Card(I). \quad (7)$$

The partition *P* is defined in a way that separates the zones where the designer has a complete certainty of the output from the zones where the certainty is only partial.

This is done by defining *P* from *AI*, because the Activation Intervals, for each dimension, separate the kernels of each label, the zones where the designer has a complete certainty, from the transition of the labels, where the certainty is only partial.

Example:

$FS=(T_n,T_c,PO,AO,D,R,MF,MF_o,I,O)$ is a fuzzy system with the membership functions defined as seen in *Fig. 2*. The set of *MF* defined are $\{A_{11},A_{12},A_{13}\}$ and $\{A_{21},A_{22},A_{23}\}$, defined respectively in I_1 and I_2. AI_1 and AI_2 will be (*Fig. 2*):

$AI_1=\{[0,A),[A,B),[B,C),[C,D),[D,E]\}$,

$AI_2=\{[0,A'),[A',B'),[B',C'),[C',D')[D',E']\}$.

P can be calculated with AI_1 and AI_2 (*Fig. 2*):

$P=\{\{[0,A),[0,A')\},\{[0,A),[A',B')\},\{[0,A),[B',C')\}...\}$.

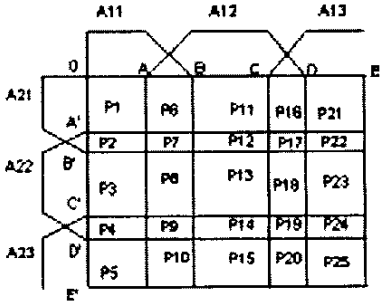


Fig.2: Representation of *MF*, *AI and P* of *FS*

■

The compiler is based on the value of the fuzzy system *FS* in the set of vertex that define *P*. For that, the matrix *V* is defined as a matrix that contains the set of vertex of partition *P*. *V* can be obtained from the Activation Points as:

$$V = \times AP_i,\ i=1,...,Card(I). \quad (8)$$

The Characteristic Matrix *CM* contains the value of the fuzzy system *FS* in each vertex of the partition *P*. *CM* is defined as:

$$CM=FS(V), \qquad (9)$$

where *FS(V)* represents the value of each element of matrix *V* in the fuzzy system given by $FS=(T_n,T_c,PO,AO,D,R,MF,MF_o,I,O)$.

Example:

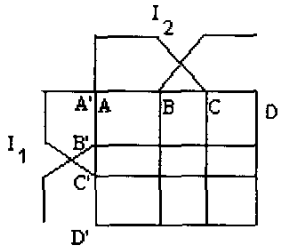


Fig. 3 Partition *P* of the fuzzy system *FS*

Given *FS* a bidimensional fuzzy system (Fig. 3), and *P* the partition of the input space, *V* and *CM* can be obtained as:

$$V=\begin{pmatrix} (A',A) & (A',B) & (A',C) & (A',D) \\ (B',A) & (B',B) & (B',C) & (B',D) \\ (C',A) & (C',B) & (C',C) & (C',D) \\ (D',A) & (D',B) & (D',C) & (D',D) \end{pmatrix}$$

$$CM=\begin{pmatrix} FS(A',A) & FS(A',B) & FS(A'.C) & FS(A',D) \\ FS(B',A) & FS(B',B) & FS(B',C) & FS(B',D) \\ FS(C',A) & FS(C',B) & FS(C',C) & FS(C',D) \\ FS(D',A) & FS(D',B) & FS(D',C) & FS(D',D) \end{pmatrix}$$

■

B. Construction of the Approximate Takagi-Sugeno System

From the information obtained in the previous point is possible to construct the equivalent Takagi-Sugeno system of a universal fuzzy system.

The Characteristic Matrix *CM* and the set of Activation Points of each dimension AP_i have the necessary information to construct the equivalent Takagi-Sugeno system.

The set of membership functions of the equivalent Takagi-Sugeno system is obtained from the set of Activation Points AP_i that define the partition *P*. This membership functions are constructed as triangular membership functions with an overlap factor of two.

Each triangular membership function of each dimension is constructed using three contiguous Activation Points. In the first and the third one the degree of truth is zero, and in the second one the degree of truth is one. Between the first and the second points the degree of truth increases linearly, and between the second and the third decreases linearly.

Fig. 4 shows an example on how from a system with trapezoidal membership functions (left), the set of membership functions that define the compiled Takagi-Sugeno system (right) is defined.

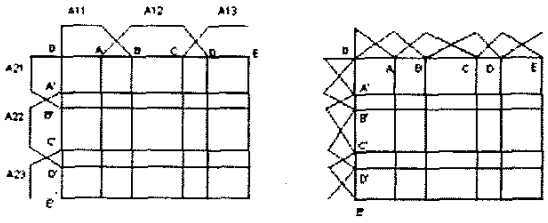


Fig. 4 Construction of the trianglar membership functions of the equivalent Takagi-Sugeno system.

The construction of the rules of the system can be done for each cell. Fig. 5 shows a cell of the partition *P* with the membership functions of the equivalent Takagi-Sugeno system. Each vertex of the cell has a value on the Characteristic Matrix *CM*.

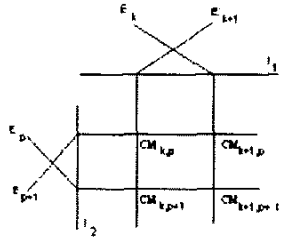


Fig. 5 Cell of the system with the membership functions.

Each vertex will define a rule of the equivalent T-S system where the antecedent will be defined by the membership functions that have a degree of truth of one in the Activation Points that define that vertex. The singleton of the output will be given by the value on the Characteristic Matrix *CM* on that vertex.

From the Fig. 5, the set of rules defined are:

$$IF\ I_1\ IS\ E_k\ AND\ I_2\ IS\ E_p\ THEN\ Z=CM_{k,p}$$
$$IF\ I_1\ IS\ E_{k+1}\ AND\ I_2\ IS\ E_p\ THEN\ Z=CM_{k+1,p}$$
$$IF\ I_1\ IS\ E_k\ AND\ I_2\ IS\ E_{p+1}\ THEN\ Z=CM_{k,p+1}$$
$$IF\ I_1\ IS\ E_{k+1}\ AND\ I_2\ IS\ E_{p+1}\ THEN\ Z=CM_{k+1,p+1}$$

A generalization of this idea will produce the set of rules of the equivalent Takagi-Sugeno system. For each vertex defined by partition *P*, a rule will be defined.

Formally, given *FS*, a universal fuzzy system, defined by the vector $FS=(T_n,T_c,PO,AO,D,R,MF,MF_o,I,O)$, the result of the compilation will be a Takagi-Sugeno system defined by a vector $FS'=(T'_n,T'_c,PO',AO',D',R',MF',MF'_o,I',O')$, where T'_n is product, T'_c is sum, PO' is prodcut, AO' is sum, D' is weighted average, $I'=I$ and $O'=O$.

The set of membership functions of each dimension of the Takagi-Sugeno system, MF'_i, with i=1...Card(I'), will be defined by the Activation Points of that dimension, AP_i.

Given n=Card(I), the algorithm to construct these membership functions is:

For i=1...n
  For j=0...Card(AP_i)-1
    $MF'_{i,j}$=TRIANGLE($AP_{i,j-1}$,$AP_{i,j}$,$AP_{i,j+1}$)
  End_For
End_For

Where TRIANGLE is a function that defines a triangle membership function using three continuous points, the first and the third define the base, and the second the point where the value of the membership function is one.

The set of rules R can be constructed with the algorithm:

For i=1...Card(MF'_1)
  For j=1...Card(MF'_2)
    ...
    For k=1...Card(MF'_n)

    R=R∪(IF I_1 IS $MF'_{1,i}$ AND ... I_n IS $MF'_{n,k}$ THEN O'=$CM_{i,...,k}$)
    End_For
    ...
  End_For
End_For

This algorithm construct the set of rules of the equivalent system obtaining the rule that each element of the Characteristic Matrix CM produces.

The set of membership functions of the output is defined as a vector that contains as singletons the elements of the Characteristic Matrix CM. MF'_o can be obtained with the algorithm:

For i=0...Card(AP_1)-1
  For j=0...Card(AP_2)-1
    ...
    For k=0...Card(AP_n)-1
      MF'_o=MF'_o∪$CM_{i,j,...,k}$
    End_For
    ...
  End_For
End_For

Those three algorithms implement the second phase of the approximate compiler, and generate the equivalent approximate Takagi-Sugeno system.

The equivalent Takagi-Sugeno system has been defined with triangular membership functions with an overlap factor of two, singletons, and prod-sum as inference system, and, as proved in [10], those kind of systems, among others, are universal appoximators.

The approximate compiler is based on a universal approximator, a Takagi-Sugeno system, and on choosing the points that construct the Takagi-Sugeno system in a way that it keeps the certainty of the designer. The equivalent Takagi-Sugeno system will act as a linear interpolator between those points [5].

C. Compilation Example

FS is a fuzzy system with two inputs, I=(I_0,I_1), and one output O, the T-norm, T_n, is *min*, the T-conorm, T_c, is *max*, the propagation operator, PO, is *min*, the aggregation operator, AO, *max* and the defuzzification algorithm is center of gravity. The system is defined also by a set of rules R and by set of trapezoidal membership functions both on the input, MF, and the output MF_o. The output produced by the system is presented in Fig. 6.

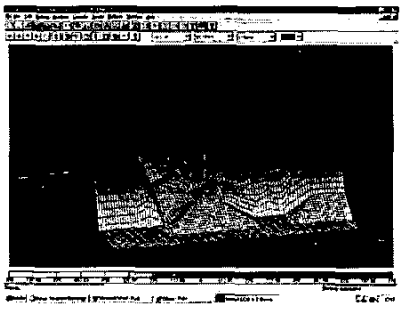


Fig. 6 Surface produced by FS.

The compilation of the system FS using the compiler defined in the previous point produces an equivalent approximated Takagi-Sugeno fuzzy system FS'. The output of FS' is presented on Fig. 7.

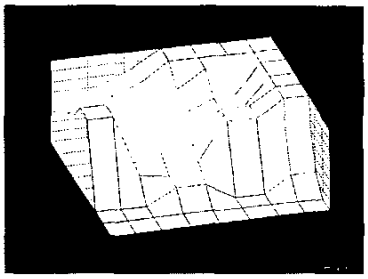


Fig. 7 Surface produced by FS'.

Comparing the value of 48682 points of each system, 35% of this points have the same value, and 65% of the points have an error inferior to ten units. The average error is of 11 units. Being the range of the output 1900 units, the error of the output is in average 0.77% of the range of the output.

As it can be seen, both systems present the same behavior. This is possible because the approximate compiler is defined in a way that it captures the certainty of the designer. The points where the designer of the fuzzy system has a complete

certainty are the ones captured by the kernel of the membership functions, and this points keep their value. The points where the designer has only a partial certainty are approximated, but in this case, the error is not important because this points are constructed also as an approximation, the designer has no clear idea of what is the value of the output.

D. Implementation

The Takagi-Sugeno system produced by the compiler has been implemented as a multilinear interpolator using C in a Texas Instruments DSP, the TMS320C6201[11].

The speed has been tested with a 2 input / 1 Output system, a 3 Input / 1 Output system and a 4 Input / 1 Output system. The results obtained can be seen in *Table I.*

TABLE I. EXECUTION TIME ON A C6201

System	Inference Time (ns)	MFLIPS
2 I / 1 O	220 ns	4.5
3 I / 1 O	512 ns	1.8
4 I / 1 O	1055 ns	0.95

The results given in *Table I* are completely satisfactory, and can be compared with the results obtained with others architectures. *Table II* gives the time response of some fuzzy coprocessors to compare the speed obtained.

TABLE II
PROCESSING TIME OF SOME FUZZY COPROCESSORS

System	FZP-0401A	WARP 2.0	SAE81C99
2 I / 1 O	0.48 MFLIPS	----------	----------
3 I / 1 O	----------	----------	12 µs
4 I / 2 O	----------	33.1 µs	----------

FZP-0401A[12] is an ASIC that processes fuzzy systems also using interpolation. WARP 2.0[13] and SAE81C99[14] are fuzzy commercial coprocessors of ST Microelectronics and Siemens respectively.

The conclusion is that standard architectures can process fuzzy knowledge even faster than specialized coprocessors using a compilation that adapts the knowledge to the standard architecture.

IV. CONCLUSIONS

An approximate compiler that transforms a universal fuzzy system into a Takagi-Sugeno fuzzy system has been proposed.

The result of the compilation is not exactly the original fuzzy system, but a system, expressed as a Takagi-Sugeno system, that respects the certainty of the designer. The certainty of the designer is captured using the kernels of the defined membership functions. This compiler allows to execute any fuzzy system in the set of platforms (both software and hardware) designed to execute Takagi-Sugeno systems.

REFERENCES

[1] C. Von Altrock, *Fuzzy Logic&Neurofuzzy Applications in Busines*, Prentice-Hall 1997

[2] H. Eichfeld, "A 12b General-Purpose Fuzzy Logic Controller Chip", *IEEE Transactions on Fuzzy Systems*, Vol 4, No. 4:460-475 ,1996

[3] N. Yubazaki, "Fuzzy inference chip ZP-0401 based on interpolation", *Fuzzy Sets and Systems*, Vol.98(3):299-310

[4] T. Takagi, M. Sugeno, "Fuzzy identification of systems and its application to modelling and control", *IEEE Transactions on Systems, Man and Cibernetics*, vol. 15, no. 1, pp. 116-132, Feb. 1985

[5] A. Jiménez, F. Matía, "A Fast Picewise-Linear Implementation of Fuzzy Controllers", *Proc. of the 1994 First International Joint Conference of NAFIPS-IFIS-NASA*, San Antonio, TX, Dic. 1994, pp. 27-31

[6] J. Gebhardt and C. Von Altrock, "Recent Successful Fuzzy Logic Applications in Industrial Automation", *Proceedings of the 5th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 96*, New Orleans, Sep. 1996

[7] G. Zahlman and M. Scherf, "Monitoring Glaucoma by Means of a NeuroFuzzy Classifier", *Fuzzy Logic Aplication Note series*, Inform Corp, http://www.fuzzytech.com, 1998

[8] M. Jamshidi, A. Titli, L.A. Zadeh, S. Boverie, *Applications of Fuzzy Logic. Towards a High Machine Intelligence Quotient Systems*, Prentice-Hall, 1997

[9] D. Elting, M. Fennich, R. Kowalczyk, B Hellenthal, "Fuzzy Anti-Lock Brake System Solution", *Intel Corporation's Automotive Operation Center Reports*, http://developer.intel.com/design/mcs96/designex/2351.htm, 1998

[10] J.L. Castro, "Fuzzy Logic Controllers Are Universal Approximators", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, No. 4, Abr. 1995, pp. 629-635

[11] Texas Instruments, "TMS320C62x/C67x CPU and Instruction Set. Reference Guide - SPRU 189C", March 1998

[12] N. Yubazaki, "Fuzzy inference chip ZP-0401 based on interpolation", *Fuzzy Sets and Systems*, Vol.98(3):299-310

[13] SGS-Thomson Microelectronics, "Weight Associative Rule Processor WARP 2.0" ,1994

[14] H. Eichfeld, "A 12b General-Purpose Fuzzy Logic Controller Chip", *IEEE Transactions on Fuzzy Systems*, Vol 4, No. 4:460-475 ,1996